基于卷积神经网络和 Tree-LSTM 的微博情感分析 *

王文凯，王黎明，柴玉梅

(郑州大学 信息工程学院，郑州 450001)

摘　要：微博情感分析旨在研究用户关于热点事件的情感观点，研究表明深度学习在微博情感分析上具有可行性。针对传统卷积神经网络进行微博情感分析时忽略了非连续词之间的相关性，为此将注意力机制应用到卷积神经网络（CNN）模型的输入端以改善此问题。由于中文微博属于短文本范畴，卷积神经网络前向传播过程中池化层特征选择存在丢失过多语义特征的可能性，为此在卷积神经网络的输出端融入树型的长短期记忆神经网络（LSTM），通过添加句子结构特征加强深层语义学习。在两种改进基础上构造出一种微博情感分析模型（Att-CTL），实验表明该模型在微博情感分析上具有优良的特性，尤其在极性转移方面仍保持较高的 F1 值。

关键词：卷积神经网络；注意力机制；长短期记忆神经网络；微博情感分析

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2017.11.0735

Sentiment analysis of micro-blog based on CNN and Tree-LSTM

Wang Wenkai, Wang Liming, Chai Yumei

(*School of Information Engineering*, *Zhengzhou University* 450001, *China*)

Abstract: Micro-blog sentiment analysis aims to study the emotional views of users on hot events, and research shows that deep learning is feasible in micro-blog's sentiment analysis. In view of traditional convolutional neural network, micro-blog sentiment analysis ignores the correlation between discontinuous words. Therefore, this paper applied attention mechanism to the input end of convolutional neural network (CNN) model to improve this problem. Because Chinese micro-blog belongs to the short text category, there was a possibility of losing too many semantic features in the selection of pooling layer features in the process of convolutional neural network forward propagation, so into the long short term memory neural network tree at the output of the convolutional neural network terminal (LSTM) , by adding the sentence structure to strengthen the deep semantic learning. Based on the two improvements, it constructed a Chinees micro-blog sentiment analysis model (Att-CTL) . Experiments show that the model has excellent characteristics in Chinese micro-blog sentiment analysis, especially in polarity shifting, and maintains a high F1 value.

Key words: CNN; attention mechanism; LSTM; micro-blog sentiment analysis

0　引言

随着社交网络的快速发展，用户可以运用微博、博客、社区等平台来抒发自己的情感，通过微博、博客等分析社交网络中用户的情感倾向引起了学术界的广泛关注[1]。

情感分析又称倾向性分析和意见挖掘，对主观性文本进行分析、处理、归纳和推理的过程。目前情感分析的主要研究方法大多是基于机器学习的传统算法，但是机器学习算法[2]借助于大量人工标注的特征来确定给定文本的情感极性。机器学习算法虽然性能优越，但需要大量的人工和领域知识，特征扩展性不灵活，而从大量训练数据中主动学习特征的深度学习方法较为适用。

卷积神经网络（CNN）是主流的深度学习模型，已很好地运用到文本分类上[3~5]。传统卷积神经网络进行文本建模时，滑动窗口对微博句子卷积操作，提取的特征是局部相连词之间的特征，忽略非相连词之间的长距离相关性。研究发现在微博文本中存在极性转移，张小倩[6]对极性转移现象进行了研究分析，将其大致分为强调、否定、转折三类。极性转移现象与非连续词的情感相关性密切相连。这种非连续词之间的相关性是传统卷积神经网络最大的限制。本文采用注意力机制来解决这个问题，通过注意力机制可以捕获长距离上下文之间的相关性。关于注意力机制在文章的第 3 章将重点介阐述。微博多为短文本，含有较少的特征信息，卷积神经网络在池化层进行特征选择时可能丢失较多的信息特征影响情感分析的准确性，并且卷积神

收稿日期：2017-11-05；**修回日期：**2017-12-27　　**基金项目：**社会媒体文本情感可视计算方法研究基金项目（U1636111）

作者简介：王文凯（1990-），男，硕士，主要研究方向为机器学习、自然语言处理（pywkwang@163.com）；王黎明（1963-），男，教授，主要研究方向为现代软件工程技术、分布式人工智能、数据挖掘等；柴玉梅（1964-），女，教授，主要研究方向为机器学习、自然语言处理.

经网络模型是将文本看做有序列的词语组合，仅考虑了文本的有序性信息而忽略了文本在语义上的结构性。结构化模型则试图将文本看做有结构的词语组合，使得在学习文本特征时能够充分保存文本的结构特征。Tai 等人[7]利用 LSTM 模型试以一种树型结构来描述文本，该模型在情感分析上取得了不错的效果。本文也采用树型的 LSTM 网模型，以提取句子的结构特征。

总结以上分析，本文在卷积神经网络模型的基础上，在模型输入端添加注意力机制，在模型输出端融合树型结构的 LSTM，得到最终的情感分析模型。通过在公开数据集上的实验，与现有结果相比，证明本文所构建模型的有效性。

1　相关工作

1.1　神经网络

随着深度神经网络在文本建模上的不断深入，研究主要集中在卷积神经网络和长短期记忆神经网络上。Kim[3]在 2014 年将卷积神经网络成功地应用在句子建模上。Kalchbrenner 等人[4]对卷积神经网络的结构进行优化，采用两个卷积层，并提出一种新的动态池化策略捕获句子内长距离词之间的相关性，以此提高情感语义的合成性能。Zhou 等人[5]结合卷积神经网络和长短期记忆神经网络，用卷积操作提取序列短语特征做为 LSTM 的输入最终获得句子表示。Zhou 等人[8]先用 LSTM 学习句子的历史信息，然后将每一时刻的输出作为卷积神经网络的输入，不同的是其 CNN 采用二维的方式进行卷积核池化操作。文献[5,8]只是将卷积神经网络和长短期记忆神经经网络以链式的拼接，在一定程度上提高了句子建模的效果。以上这些方法以自主学习的方式为语言建模更为本质地刻画语言特征，从而能提高利用机器学习方法解决这些传统任务时的性能。

1.2　情感分析

微博情感分析通常采用文本情感分析[9]的相关技术。Pang 和 Lee 等人在 2008 年就利用词袋模型对文本进行情感分析，随后许多人尝试设计更好的工程特征或者使用基于句法结构的极性转移规则来提升情感分析的准确率。这些模型都是基于词袋模型，无法获取到文本中的深层语义信息，因此效果并不理想。Santos 等人[10]提出了 CharsCNN 模型，将两个卷积层分别学习词语的构造特征和句子的语义特征，进行情感分析取得较好结果。Bahdanau 等人[11]首次将注意力机制用到自然语言处理方面，处理源语言与当前目标语言之间的关联性。本文受他们启发将注意力机制引用到卷积神经网络模型的输入端，以解决情感分析中非连续词相关性问题。Socher 等人[12~14]利用改进的递归模型获取句子语义特征，处理情感分析问题中语义合成的问题。考虑到递归神经网络独特的优势，本文另辟蹊径将递归神经网络最终的分布式表示融入卷积神经网络的输出端。

2　混合神经的微博情感分析算法

2.1　混合神经网络模型构造

深度学习的方法可通过自我学习的方式来学习到微博中内在的情感语义特征。本文深度神经网络模型（图 1）的构造从以三个方面考虑：a）通过在卷积神经网络模型的输入端引入注意力机制学习非连续词之间的语义相关特征；b）在卷积神经网络的输出端融入长短期记忆神经网络学习到的句子结构特征；c）卷积神经网络和长短期记忆神经网络以联合训练的方式学习句子内语义信息和结构特征。

如图 1 所示，模型以微博句子的分布式词向量作为模型的原始输入；模型左边部分为长短期记忆神经网络的结构语义学习，其输出为 h_t；模型右边部分为卷积神经网络模型潜在特征学习，其输出为 h_p，最后联合语义学习隐藏层作为分类器的输入进行情感分类。

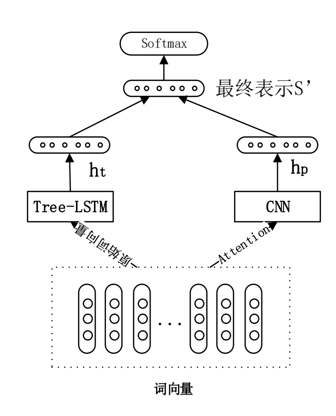


图 1　注意力机制的卷积长短期神经网络（Att-CTL）

Att-CTL 的模型描述和情感分析算法如下：

输入：训练语料 X 和对应标签 Y。

数据预处理：截取微博为相同长度，句子词较少的用 0 填充。

初始化所有模型参数为小的随机值。

在遇到终止条件前：

对于训练样例中的每个< $\vec{X},\vec{y}$ >：

1.把输入沿网络前向传播：

CNN 卷积池化后的最终句子表示：　h_p

TreeLSTM 的根节点的隐藏输出：　h_{root}

最终句子特征表示：　$S = h_p + h_{root}$

Softmax 层微博情感极性的概率计算：

$$p(y\,|\,P) = soft\max(w_s \cdot S + b_s)$$

代价函数：　$J(w,b)$

2.反向传播更新模型权值 w,b

$$w \leftarrow w + \Delta w\ ,\ b \leftarrow b + \Delta b$$

其中 $\Delta w = \dfrac{\partial J(w,b)}{w}, \Delta b = \dfrac{\partial J(w,b)}{b}$

2.1.1 基于注意力机制组合词向量特征

为获得词语的语义信息，通过 skip-gram 模型[15]来训练词向量。其中 skip-gram 模型原理是根据给由给定中心词预测周围的词，假设句子中的词组序列为 $w_1, w_2, \ldots, w_n$，模型通过训练使式中（1）的条件概率最化：

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{-c\le i\le c, i\ne 0}\log p(w_{n+i}\,|\,w_n) \tag{1}$$

其中：c 是当前词语的上下词数，c 值越大训练样例越多训练效果越好，但训练时间相应增加。

在微博句子中的一些否定词、程度副词、转折词等的使用可能导致句子的情感极性偏移。这种情感极性转移现象与非连续词的情感相关性密切相连。针对微博的这种语言现象，在 skip-gram 模型训练的词向量基础上采用注意力机制重组词向量。

注意力机制层引入到原始词向量输入层与卷积层之间，其具体实现如图 2 所示。注意力机制层是为每一个词创建一个上下文向量。上下文向量连接词向量组合成一个新的词向量表示，它将最为卷积神经网络卷积层的输入。直观地说，一对彼此相距甚远的词往往有较小的相关性。因此，该模型将距离衰减应用到注意力机制上。

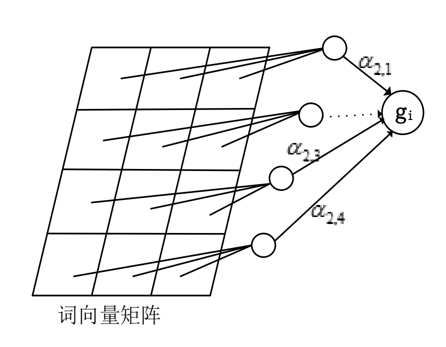


图 2　重组上下文词向量

构建图 2 中词向量矩阵 $X \in R^{l \times d}$，词向量矩阵的表示：$X = [x_1, x_2, \cdots, x_l]$，$x_i \in R^d$ 其中 l 是给定句子的长度，d 是实数词向量的纬度。注意力机制的思想是在生成 x_i 的上下文向量 g_i 时，将注意力集中在特定的有意义的词上。这种机制决定了在预测句子情感类型时要注意哪些词比句子中的其他词更应该受关注。包含上下文信息的词向量表示为

$$g_i = \sum_{j \neq i, \cdots i+n} \alpha_{i,j} \cdot x_j \tag{2}$$

其中：$\alpha_{i,j}$ 是注意力机制权重，要求 $\alpha_{i,j} \geq 0$，并且通过 SoftMax 正则化技术使 $\sum_j \alpha_{i,j} = 1$。注意力机制权重的计算方程为

$$\alpha_{i,j} = \frac{e^{score(x_i, x_j)}}{\sum_{j'} e^{score(x_i, x_{j'})}} \tag{3}$$

$$score(x_i, x_j) = v_a^T \tanh(w_a[x_i \oplus x_j]) \tag{4}$$

其中 $score(x_i, x_j)$ 就是通过 MLP 实现的，用 MLP 来建模词对 $(x_i, x_{j,\ i \neq i})$ 的相关性。

情感分类实例：这部电电影中没有一个像样的表演，也没有一行巧妙的对白。

在这句话中，学习词“表演”的上下文向量时，注意机制比其他词更注重“没有”和“像样的”。score(表演，没有)和 score（表演，像样的）比其他的词对值更大。

最后，本文定义扩展词向量 x_i'，x_i' 由 x_i 和它的上下文向量 g_i 的拼接，然后用 $X' = [x_1', x_2, ..., x_l']$ 作为新的句子矩阵，x_i' 将被作为 2.1.2 中的 CNN 的输入。

$$x_i' = x_i \oplus g_i \tag{5}$$

2.1.2 卷积神经网络

卷积神经网络可分为四层，依次为输入层、卷积层、池化层、情感分类输出层。

输入层：由注意力机制得到的重组词向量矩阵 S' 作为卷积神经网络的输入，$S' = [x_1', x_2, ..., x_l']$。$x_i' \in R^d$ 是句子中对应的第 i 个的词向量表示，由于句子的长度为 l，所以最终的句子表示为 $S' \in R^{l \times 2d}$。

卷积层：卷积层通过指定不同的窗口值提取句子的不同的特征信息。在卷积层通过卷积核来抽取句子的特征表示 c，本文设定卷积的上下文窗口的宽度为 n，卷积的权重矩阵的 $W \in R^{n \times 2d}$，卷积后获得句子的特征映射为

$$c_i = g(W \cdot x_{i:i+n-1}' + b) \tag{6}$$

其中：$1 \leq i \leq l-n+1$；c_i 是特征映射中第 i 个特征值；g 是非线性激活函数，本文卷积核函数采用 Rectified liner 函数；b 是偏置量。为了抽取句子多个特征，在卷积中使用多个过滤器。本文设定 m 个过滤器，卷积操作如式（7）所示。

$$c_{ji} = g(W \cdot x_{i:i+n-1}' + b) \tag{7}$$

其中：$1 \leq j \leq m$；卷积操作的最终结果为矩阵 $C \in R^{m \times (l-n+1)}$。

池化层：为了提取特征映射中最重要的特征，卷积神经网络对每一个特征映射抽取到的特征值，只取其中权重最大的作为池化层的的保留值，其他的特征值全部舍弃。神经元连接如图 3 所示。通过池化层得到 p_{ji}，其计算公式为

$$1 \leq j \leq m \tag{8}$$

$1 \leq j \leq m$，$1 \leq i \leq l-n+1$。然后拼接所有的 p_{ji} 组合成向量 $p \in R^m$。其中:m 提取的特征个数;P 代表句子级特征。

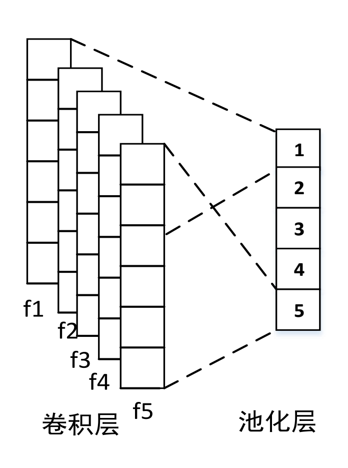


图 3　最大值提取的池化操作

输出层：池化层获取的特征向量 P 作为输出层的输入。输出层选择 softmax 回归函数。为了避免过拟合在 softmax 之前添加 Dropout 层，其思想是按照一定的概率屏蔽神经网络单元。具体计算如下：

$$p(y \mid P, w_s, b_s, q) = soft\max(w_s(P \circ q) + b_s) \tag{9}$$

其中：$w_s \in R^{|S| \times |c|}$ 和 $b_s \in R^{|c|}$ 是 softmax 层的参数；q 是 Dropout 层的屏蔽向量；$\circ$ 表示元素相乘；$|c|$ 为情感类别数。

2.1.3 Tree-LSTM 结构语义信息提取

在 2.1.2 节卷积神经网络池化层中的池化策略为 Max Pooling，该操作保证特征的旋转不变性，不论这个重要特征在哪个位置出现，都会不考虑其出现位置将其提取出来。这对图像处理是很好的特性，但对于自然语言处理，特征出现的位置很重要，例如主语一般出现在句首，宾语一般出现在句尾等。这些位置信息在分类任务里占有很大比重，但 Max Pooling 把这些信息丢弃。为了弥补这个缺陷，本文引入树型 LSTM。Tai[7] 等人对树型的 LSTM 结构作了深入的剖析，它根据句子语法的树型结构由底向上递归地合并相邻的节点。详细的 Tree-LSTM 结构如图 4 所示。

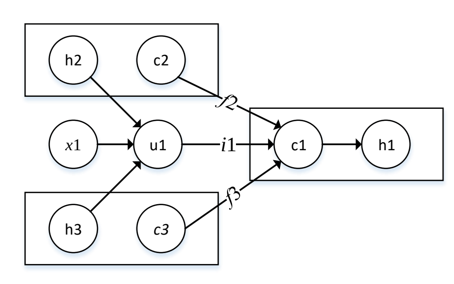


图 4 树型结构的 LSTM 网络

Tree-LSTM 中第 j 个几点包含：记忆单元 c_j 、隐藏单元 h_j 、输入门 i_j 和输出门 o_i 。Tree-LSTM 根据语法分析树结构构建，每个单元依赖其他多个孩子，如图 3 所示，单元 1 的 c_1 值依赖于其他连个孩子的 c_2 和 c_3 。对于任意一个孩子节点 k ，单元 j 都有一个对应的遗忘门 f_{jk} 。对于任意一个二元 Tree-LSTM 单元 j ，c_{jk} 、h_{jk} 分别是节点 k 的记忆单元和隐含状态。构建 Tree-LSTM 网络采用的是句子成分树，所以 k 取值为 1 或 2。Tree-LSTM 神经网络前馈过程为

$$i_{\ j}=\sigma(W^{i}x_{j}+\sum_{k=1}^{2}U_{k}^{(i)}h_{jk}+b^{(i)}) \tag{10}$$

$$f_{jk}=\sigma(W^{f}x_{j}+\sum_{l=1}^{2}U_{kl}^{(i)}h_{jl}+b^{(f)}) \tag{11}$$

$$o_{j}=\sigma(W^{o}x_{j}+\sum_{k=1}^{2}U_{k}^{(o)}h_{jk}+b^{(o)}) \tag{12}$$

$$u_{j}=\tanh(W^{u}x_{j}+\sum_{k=1}^{2}U_{k}^{(u)}h_{jk}+b^{(u)}) \tag{13}$$

$$c_{j}=i_{j}\otimes u_{j}+\sum_{k=1}^{2}f_{jk}\otimes c_{jk} \tag{14}$$

$$h_{j}=o_{j}\otimes\tanh(c_{j}) \tag{15}$$

其中：σ 是 sigmoid 函数；b 是偏置项；U 表示孩子节点隐含值得权重；l 表示第 k 个孩子的节点的第 l 个孩子节点；W 表示不同结构内的权重；$\otimes$ 表示向量对应元素相乘。计算模型中的任意一单元，其孩子节点采用不同的参数矩阵。Tree-LSTM 模型通过逐步训练将树根节点的隐含输出 h_{root} 作为句子的向量表示。将 Tree-LSTM 的输出的句子表示与卷积神经网络的池化层的句子表示拼接在一起，得到句子的最终表示 $S=[h_p,h_{root}]$，将句子的最终表示作为卷积神经网络 softmax 层的输入。

2.2 混合神经网络模型训练

本文模型属于有监督模型，模型根据输出的结果式（9）计算残差并更新模型参数。模型更新的参数包括卷积神经网络和 Tree-LSTM 的参数。假设训练集的样本由 m 个已标记的样本构成：$\{(x^1,y^1),\ldots,(x^m,y^m)\}$（符号约定，$x$ 的维度为 n+1,其中 x_0 =1 为截距项）。以 x 为输入，通过 softmax 计算得到句子的情感标签和模型参数集 $\theta=\{\theta_1,\ldots,\theta_k\in R^{n+1}\}$ 的条件概率分布，计算如下：

$$p(y^{i}=j\mid x^{i},\theta)=\frac{e^{\theta_{j}^{T}x^{i}}}{\sum_{l=1}^{k}e^{\theta_{j}^{T}x^{i}}} \tag{16}$$

其中：k 为标签数量（本文仅关注负向和正向两类，即 k=2）。

接下来训练模型参数 θ，训练时采用交叉熵作为损失函数，并且引入权重衰减（weight decay）对参数进行正则化。具体计算如下所示：

$$J(\theta)=-\frac{1}{|m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y^{i}=j\}\log p(y^{i}=j\mid x^{i},\theta)\right]+\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=1}^{n}\theta^{2} \tag{17}$$

其中：y_c 为句子的真实情感值；$p(c\mid x,\theta)$ 为预测的情感值；$1\{\cdot\}$ 是实行函数，其取值规则为：$1\{True\}=1$，n 表示参数 θ 的数量；λ 是正则化系数，目的是防止过拟合提高泛化能力。

由于 $\lambda>0$，损失函数变成严格的凸函数，这样就保证了解的唯一性。本文不模型训练采用小批量梯度下降法训练法最小化损失函数 $J(\theta)$，即每次取整个样本集的一部分计算梯度更新权值。当然也可以采用 AdsGrad 或 LBFFGS[16,17]等方法优化参数。更新参数需要对 $J(\theta)$ 求导，具体计算如下：

$$\nabla_{\theta_{j}}J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\left[x^{i}(1\{y^{i}=j\}-p(y^{i}=j\mid x^{i};\theta))\right] \tag{18}$$

梯度下降算法依据偏导式（18）最小化 $J(\theta)$ 。参数 θ 的更新方式如下：

$$\theta_{j}\leftarrow\theta_{j}-\alpha\nabla_{\theta_{j}}J(\theta)(j=1,\ldots,k) \tag{19}$$

其中：α 为学习率。

3 实验

3.1 实验环境

本文的实验环境及其配置如表 1 所示。

表 1 实验环境配置

实验环境	环境配置
操作系统	Windows 7
处理器	AMD Athlon X4 750
内存	4GB
编程语言	Python 2.7
深度学习框架	Theano 0.9

3.2 实验数据集

本文模型属于有监督的深度学习模型，需要较多高质量的

训练数据才能保证模型的有效性。本文采用 2014 年中文观点倾向性分析评测微博数据集，数据集分为训练数据和测试数据集。训练数据集包括官方公布和手工标注总共 8 017 条微博，其中积极数据 4 281 条，消极数据 3 736 条。测试数据选不同主题微博数据 2 317，其中积极数据 1 204 条，消极 1 113 条。通过多次重复实验评估各个模型的性能，实验的平均值作为最终结果。

为了突出 Att-CTL 模型的性能，本文还将选取数据集 5 317 条，包含情感极性转移的数据 2 159 条和不包含情感极性转移数据 3 158 条，分别在 W2V+CNN、C-LSTM 模型和 Att-CTL 模型上实验。

3.3　实验可调参数设置

Zhang 等人[18]在句子分类的任务对 CNN 进行了敏感性分析，本文参照其建议设置本文模型参数。模型中的参数通参数主要有词向量的维度 d、卷积核滑动窗口大小 n、卷积核数量 m、Dropout 算法的比率 ρ。本文选择 Word2vec 工具中的 skip-gram 模型，它由 2 000 万条的微博语料训练产生，d 的取值为{50,100,150,}。关于 CNN 框架中 n 的取值组合设置为(2,3,4)、(3,4,5)和(4,5,6)，m 在三种卷积操作的三个通道中保持不变，取值为{100,200,300}。d、n 和 m 三种参数构成了 27 种组合，实验通过网格搜索的方法调参。Tree-LSTM 的输出维度与 CNN 的特征映射数保持一致，以更好地融合成最终的句子表示，其维度设置为 300。将平均实验结果最优的参数组合作为最终结果，实验参数如表 2 所示。

表 2　模型参数设

参数名	值
词向量纬度	50
上下文窗口值	3.4,5
卷积单元个数	100,100,100
Tree-LSTM 输出维度	300
比率 ρ	0.5

实验结果发现，模型的训练的迭代次数对情感分析影响很大，迭代次数的越大模型对训练数据的拟合越好，这样会导致过拟合问题。如图 5 所示，当迭代次数为 15 次数，在训练数据和测试数据上的 F 值都能达到 85%左右，因此，实验设置训练迭代次数为 15 次。

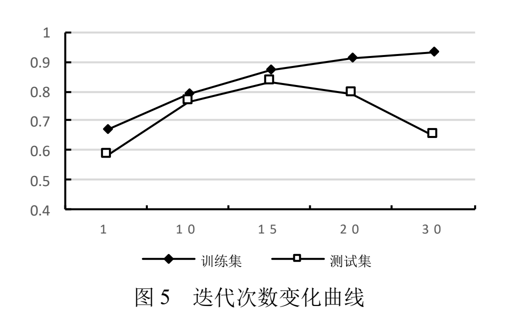


图 5　迭代次数变化曲线

3.4　本文模型与基本模型的对比

为了验证模型的有效性和准确性，设计六组实验对模型进行性能对比。

第一组：传统机器学习，在相同的数据集上，为保证实验的有效性，将基于注意力机制组合后的特征向量作为输入，分别用线性回归（linear regression）、支持向量机（SVM）方法进行情感分类来证明 Att-CTL 模型在情感分析任务上的性能优势。其中 Linear regression 是一个线性模型、SVM 构造一个超平面在情感分类任务中取得过较好的结果[19]。

第二组：Skip-gram +CNN。模型采用 skip-gram 模型训练词向量，将句子特征矩阵输入基于注意机制的卷积神经网络模型，证明添加注意力机制后模型在情感分析任务上的性能优势。

第三、四、五组将 skip-gram 模型训练的词向量作为输入。第三组：Att-CNN 模型，基于注意力机制的卷积神经网络（attention based convolution neural network，Att-CNN）在卷积神经网络模型的输入端融入注意力机制，证明在卷积神经网络模型池化层之后融入 Tree-LSTM 在提取句法结构的性能优势。第四组：C-LSTM 模型，模型来自 Zhou 等人[8]提出的基于卷积神经网络和长短期记忆神经网络的序列结构模型，证明 Att-CTL 模型中 Tree-LSTM 的融入方式在情感分析上的有效性。第五组：Att-CTL 模型，本文融合注意力机制、Tree-LSTM 进行情感分类的模型。

第六组：W2V+CNN，Kim 提出的基于 word2vec 训练的词向量的卷积神经网络模型，证明 Att-CTL 模型比传统卷积方法更胜任情感分析任务。

3.5　实验结果分析

在机器学习、自然语言处理中评估是一个重要的环节，评价指标通常采用精确率(precision)、召回率(recall)和 F1-measure。Precision 评估的是查准率，recall 评估的是查全率，F1 值是综合评价指标。六组实验在 COAE2014 微博数据集上三个指标的测评结果如表 3 所示。

表 3　不同模型的测试结果

模型	Precision	Recall	F1
Linear regression	0.5108	0.4710	0.4865
SVM	0.7172	0.7204	0.7188
Skip-gram+CNN	0.7468	0.7061	0.7258
Skip-gram+Att-CNN	0.8217	**0.7587**	0.7889
C-LSTM	0.8409	0.7511	0.7934
W2V+CNN	0.7558	0.7103	0.7323
Att-CTL（本文）	**0.8712**	0.7458	**0.8036**

实验结果表明，采用 CNN 的模型在情感分类任务上总体优于机器学习方法，Att-CTL 模型在 Precision 结果最优，较其他模型有较大提升。基于注意力机制的 Att-CNN 和 Att-CTL 模型较经典模型 W2V+CNN 在准确率和召回率上都有提升。在 F1 值上 Att-CTL 模型分别高于 C-LSTM 和 Att-CNN 模型，取

得了最好的结果。

将 Linear regression 和 SVM 方法与 W2V+CNN 模型对比。在分布式词向量特征表示的基础上，卷积神经网络模型在特征构建上优于机器学习方法，且卷积神经网络通过局部感知野和权值共享，减少了模型参数。W2V+CNN 模型在 F1 上明显高于这两种模型，证明卷积神经网络在情感分类上的出色表现。

将 Skip-gram+Att-CNN 与 Skip-gram+CNN 模型作对比，在 F1 上提升了 6.31%。以 skip-gram 模型训练的词向量作为输入，在相同的卷积神经网络条件下，采用注意力机制获得长距离上下文信息的 Att-CNN，能有效提升文本情感分析的准确率。

将 Att-CTL 与 Att-CNN 模型作对比，两者都以基于注意力机制重组的上下文词向量作为输入。将 Att-CTL 模型与 C-LSTM 模型作对比，两者以不同的组合方式获得句子表示，本文的组合方法能更准确地的进行情感分类。

为了突出本文模型的注意力机制在解决长距离的上下文相关性特性，采用句子情感极性转移的数据集测试模型性能。其实验结果如图 6～8 所示。

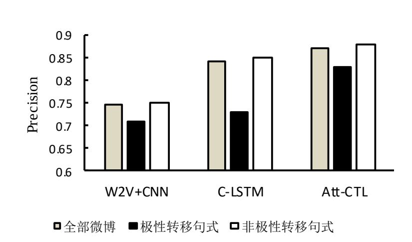


图 6　实验结果的准确率

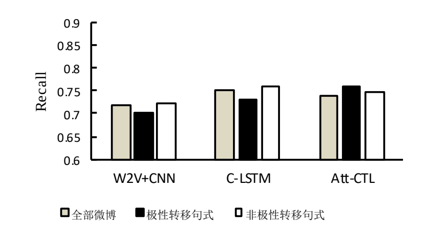


图 7　实验结果的召回率

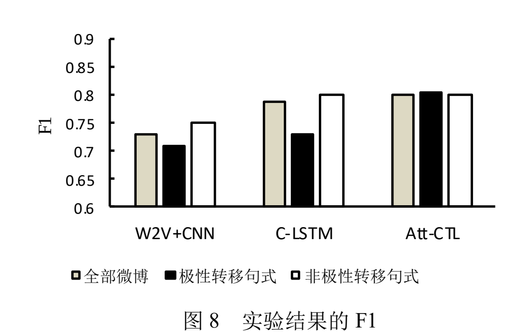


图 8　实验结果的 F1

W2V+CNN 和 C-LSTM 模型在极性转移句式的 Precision 指标都明显低于非极性转移句式，而 Att-CTL 模型在极性转移句上的性能指标有很大提升，与非极性转移句式保持很小的差别。特别从图 8 得出，在总微博数，极性转移句式和非非极性转移句式数据上的 F1 变化不大，同时也是实验的最好结果，表明 Att-CTL 模型在转折句式上准确处理情感分类任务的充分优势。

4　结束语

本文深入分析卷积神经网络的内部结构，针对微博句子存在的极性转移现象在句卷积神经网络模型的输入层之前引入注意力机制解决了非连续词之间的相关性问题。为了获得句子的结构特征，利用树型结构的 LSTM 学习文本特征并将其与卷积神经网络的池化层输出相融合组成最终的文本表示。本文模型在解决情感极性转移现象的同时，还获取了文本的结构信息，使学习到的特征更加丰富。在 COAE2014 数据集进行训练和测试，证明基于注意力机制的卷积神经网络和树型结构的长短期记忆神经网络模型能更准确地完成情感分析任务。

参考文献：

[1] Bo P, Lee L. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales [C]// Proc of the ACL. 2005: 115-124.

[2] Liu B. Sentiment analysis and opinion mining [C]// Proc of Synthesis Lectures on Human Language Technologies. 2012: 152-153.

[3] Kim Y. Convolutional neural networks for sentence classification [C]// Proc of Empirical Methods in Natural Language Proccessing. 2014: 1746-1751.

[4] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences [C]// Proc of ACL. 2014: 655-665.

[5] Zhou C, Sun C, Liu Z, *et al.* A C-LSTM neural network for text classification [J]. arXiv preprint arXiv: 1511. 08630, 2015.

[6] 张小倩. 情感极性转移现象研究及应用 [D]. 苏州: 苏州大学, 2012.

[7] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [J]. arXiv preprint arXiv: 1503. 00075, 2015.

[8] Zhou P, Qi Z, Zheng S, *et al.* Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling [C]// Proc of COLING. 2016: 3485-3495.

[9] 赵妍妍, 秦兵, 刘挺. 文本情感分析 [J]. 软件学报, 2010, 21 (8): 1834-1848.

[10] Santos C N D, Gattit M. Deep convolutional neural networks for sentiment analysis of short texts [C]// Proc of International Conference on Computational Linguistics. 2014: 69-78.

[11] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [J]. arXiv preprint arXiv: 1409. 0473, 2014.

[12] Socher R, Pennington J, Huang E H, *et al.* Semi-supervised recursive autoencoders for predicting sentiment distributions [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2011: 151-161.

[13] Socher R, Manning C D, Ng A Y. Learning continuous phrase representations and syntactic parsing with recursive neural networks [C]//

Proc of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop. 2010: 1-9.

[14] Socher R, Perelygin A, Wu J, *et al.* Recursive deep models for semantic compositionality over a sentiment treebank [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2013: 1631-1642.

[15] Mikolov T, Sutskever I, Chen K, *et al.* Distributed representations of words and phrases and their compositionality [C]// Advances in Neural Information Processing Systems. 2013: 3111-3119.

[16] Goller C, Kuchler A. Learning task-dependent distributed representations by backpropagation through structure [C]// Proc of International Conference on Neural Networks. 2002: 347-352.

[17] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization [J]. Journal of Machine Learning Research, 2011, 12 (7): 257-269.

[18] Zhang Y, Wallace B. A sensitivity analysis of (and practitioners'guide to) convolutional neural networks for sentence classification [J]. arXiv preprint arXiv: 1510. 03820, 2015.

[19] Pang B, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques [C]// Proc of the ACL-02 Conference on Empirical Methods in Natural Language Processing. 2002: 79-86.